

Pipelining and Superscalar using SimpleScalar (2)

This lab extends the previous lab on pipelining and superscalar. If you have not completed the previous lab, please do so as you won't be able to do this lab without that.

1) In this part, you will change the pipeline configuration. Every thing else remains the same as in part 1) of the previous lab.

a) For this part, make a copy of `default.cfg`, call it `config_c.cfg`. Make changes in this configuration file as in part 1) of the previous lab except that

Resources all types = 4

b) Run `sim-outorder` with the `config_c.cfg` file.

Q 1) What is the IPC (Instructions per cycle) of the `test-math` program using this configuration?

Q 2) Compare this configuration and configuration in part 1 of previous lab. Does the added resources increase performance?

2) In this part, you will change the pipeline configuration. Every thing else remains the same as in part 1) of the previous lab.

a) For this, make a copy of `default.cfg`, call it `config_d.cfg`. Make changes in this configuration file as in part 1 of **this** lab except that

Inorder = false

b) Run `sim-outorder` with the `config_d.cfg` file.

Q 1) What is the IPC (Instructions per cycle) of the `test-math` program using this configuration?

Q 2) Compare this configuration and configuration in part 1 of **this** lab. Does the out of order increase performance?

Q 3) As the system contains 4 Functional units in parallel, find the bottleneck of execution. This means changing the number of **which** parameter of these

Memory port

Issue width

Decode width

will increase the performance most. You will have to run multiple simulations to see this. Change only one of these to 4 at a time and observe the effect on IPC.

Q 4) Based on part e) what is the new IPC and what is your conclusion?

3) In this part, you will change the pipeline configuration.

a) For this, make a copy of `default.cfg`, call it `config_e.cfg`. Make the following changes in this configuration file.

```
Fetch queue size = 4,
Decode:width = 4,
Load store queue size = 8,
Register update unit (ruu) size = 8,
Issue width = 4,
Memory ports = 4,
Inorder = false,
Resources all types = 1.
```

b) Run `sim-outorder` with the `config_e.cfg` file.

Q 1) What is the IPC (Instructions per cycle) of the `test-math` program using this configuration?

Q 2) Compare this configuration with the configuration in part 2 of the previous lab.

4) In this part, you will change the pipeline configuration.

a) For this, make a copy of `default.cfg`, call it `config_f.cfg`. Every thing remains the same as in part 3 of **this** lab except
Resources all types = 4.

b) Run `sim-outorder` with the `config_f.cfg` file.

Q 1) What is the IPC (Instructions per cycle) of the `test-math` program using this configuration?

5) In this part, you will change the pipeline configuration.

a) For this, make a copy of `default.cfg`, call it `config_g.cfg`. Every thing remains the same as in part 4 of **this** lab except
Register update unit (ruu) size = 16

b) Run `sim-outorder` with the `config_g.cfg` file.

Q 1) What is the IPC (Instructions per cycle) of the **test-math** program using this configuration?

Bonus part

Plot a graph using MATLAB or EXCEL showing the performance IPC as a function of # of resources (without modifying any other parameter). For this make a copy of the configuration file **config_e.cfg** used in part 3, call it **config_bonus.cfg**

On the same graph, draw two plots. One plot should be IPC as a function of integer ALUs and the second plot should be IPC as a function of Floating point ALUs.

Modify the number of each kind of resource independently from 1 to 4 (this means if you are varying Integer ALUs, Floating point ALUs should be fixed to 1 and vice versa) and suggest which one is more useful than other in gaining performance.